

An Adaptive Finite Element Splitting Method for the Incompressible Navier–Stokes Equations

K. Selim^{a,1,2,*}, A. Logg^{b,1,2,**}, M. G. Larson^{c,3}

^aSimula Research Laboratory, P.O. Box 134, N-1325 Lysaker, Norway

^bSimula Research Laboratory, P.O. Box 134, N-1325 Lysaker, Norway

^cDepartment of Mathematics, Umeå University, SE-901 87 Umeå, Sweden

Abstract

We present an adaptive finite element method for the incompressible Navier–Stokes equations based on a standard splitting scheme (the incremental pressure correction scheme). The presented method combines the efficiency and simplicity of a splitting method with the powerful framework offered by the finite element method for error analysis and adaptivity. An *a posteriori* error estimate is derived which expresses the error in a goal functional of interest as a sum of contributions from spatial discretization, time discretization and a term that measures the deviation of the splitting scheme from a pure Galerkin scheme (the computational error). Numerical examples are presented which demonstrate the performance of the adaptive algorithm and high quality efficiency indices. It is further demonstrated that the computational error of the Navier–Stokes momentum equation is linear in the size of the time step while the computational error of the continuity equation is quadratic in the size of the time step.

Keywords: adaptive finite element method, *a posteriori* error estimate, incompressible Navier–Stokes equations, operator splitting method

1. Introduction

Adaptive finite element methods play an increasingly important role in solving complex problems in science and engineering. The adaptive methods are in general based on *a posteriori* error estimates, where the error is estimated in terms of computable quantities, and adaptive algorithms for automatic tuning of critical discretization parameters such as the time step and the local mesh size.

*Principle author

**Corresponding author

Email address: `logg@simula.no` (A. Logg)

¹Center for Biomedical Computing at Simula Research Laboratory, Norway

²Department of Informatics, University of Oslo, Norway

³Department of Mathematics, Umeå University, Sweden

Several important works have been published on *a posteriori* error estimates for finite element approximations of time-dependent problems, see for example Eriksson et al. [1], Becker and Rannacher [2], Giles and Süli [3], the research monograph Estep et al. [4] and the references therein. However, these estimates are generally restricted to finite element approximations in space and time and thus do not cover commonly used splitting schemes for efficient time stepping. Splitting schemes are used to avoid solving coupled systems of equations in each time step and have many applications, including reaction–diffusion and fluid flow problems. An *a posteriori* error estimate for a splitting method for systems of ordinary differential equations was recently presented by Estep et al. [5]. Error analysis of non-Galerkin solutions is considered by Giles and Süli [3] with particular focus on error correction; that is, improving the accuracy of a computed functional by post-processing. A *a posteriori* error analysis of the incompressible Navier–Stokes has been studied in detail before, see for example Hoffman [6], Hoffman and Johnson [7], but not for splitting methods.

In this work, we consider splitting schemes for fluid flow. More precisely, we derive an *a posteriori* error estimate for an incremental pressure correction splitting scheme for the incompressible Navier–Stokes equations. This type of scheme was originally proposed by Chorin [8] and Temam [9] and was later refined by Goda [10]. Even if a particular scheme is considered, our approach extends to other splitting schemes or any other scheme.

The basic idea is to construct a piecewise polynomial interpolation in time of the velocity and pressure and then apply the standard duality argument to derive an error representation formula. Since the splitting scheme does not satisfy a full Galerkin orthogonality, we are left with an algebraic residual measuring the effect of the lack of orthogonality caused by the splitting. The final estimate thus has three contributions measuring the effect of discretization in space, discretization in time and splitting, respectively. A similar approach was briefly proposed but not implemented or tested by Bengzon and Larson [11]. Based on the *a posteriori* error estimates, we construct an adaptive algorithm and investigate the performance of the adaptive algorithm and the quality of the error estimate.

1.1. Outline of this paper

The outline of this paper is as follows. In the next section, we present our model problem (the incompressible Navier–Stokes equations). In Section 3, we introduce the inconsistent finite element splitting method that is used to solve the Navier–Stokes equations. The *a posteriori* error analysis is presented in Section 4, and the adaptive algorithm is presented in Section 5. The efficiency of the adaptive algorithm and the quality of the error estimate is demonstrated with numerical examples in Section 6. The paper closes with a summary and some concluding remarks in Section 7.

2. The incompressible Navier–Stokes equations

We consider a fluid governed by the incompressible Navier–Stokes equations. For $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) we seek the velocity $u : \Omega \times [0, T] \rightarrow \mathbb{R}^d$ and pressure $p : \Omega \times [0, T] \rightarrow \mathbb{R}$ such that

$$\begin{aligned} \dot{u} + (u \cdot \nabla)u - \nabla \cdot \sigma(u, p) &= f & \text{in } \Omega \times (0, T], \\ \nabla \cdot u &= 0 & \text{in } \Omega \times (0, T], \end{aligned} \quad (1)$$

where f is a given body force per unit volume. In (1), the first equation is the momentum equation and the second equation is the continuity equation. The symmetric Cauchy stress tensor $\sigma(u, p)$ is defined as

$$\sigma(u, p) = 2\nu\epsilon(u) - pI, \quad (2)$$

where ν denotes the kinematic viscosity, I is the identity matrix and $\epsilon(u)$ is the symmetric gradient:

$$\epsilon(u) = \frac{1}{2}(\nabla u + (\nabla u)^\top). \quad (3)$$

We let $\Gamma = \Gamma_D \cup \Gamma_N$ denote the boundary of Ω and associate Dirichlet and Neumann boundary conditions with the two disjoint subsets Γ_D and Γ_N , respectively. On the Dirichlet boundary Γ_D , we impose a no-slip boundary condition ($u = 0$) and assume a fully developed flow at the Neumann boundary Γ_N ; that is, $\nabla u n = 0$ where n is the outward pointing unit normal. This condition ensures that the flow does not “creep around the corners” at the inflow and outflow. The boundary condition is implemented weakly by dropping the term involving ∇u from the boundary terms, leaving only $(\nu(\nabla u)^\top - pI)n$.

3. An inconsistent finite element formulation

Over the past couple of decades, numerous methods have been developed for the numerical solution of the incompressible Navier–Stokes equations. Many of these methods are based on a pseudo-compressibility in order to overcome the algebraic difficulties of solving the saddle-point problem resulting from a direct discretization of the Navier–Stokes equations. A particular type of schemes are the so-called splitting schemes where the velocity and pressure variables are computed in a sequence of predictor–corrector type steps.

The first splitting method developed for the Navier–Stokes equations is the Chorin projection scheme [8, 9]. Chorin’s scheme is a so-called non-incremental pressure correction scheme where the starting point is to neglect the pressure in the momentum equation and solve for a tentative velocity field. The tentative velocity is then projected onto a divergence free space, resulting in a Poisson problem for the pressure. In Goda [10], an improved version of Chorin’s scheme, the Incremental Pressure Correction Scheme (IPCS), was presented, which solves the incompressible Navier–Stokes equations in three steps. In the first step, an explicit pressure (the value from the previous time step) is used in the momentum equation and in the two subsequent steps, both the pressure and the velocity are corrected.

In Valen-Sendstad et al. [12], a comparison is made between different splitting schemes, including a recent scheme by Guermond and Shen [13], and (stabilized) Galerkin finite element methods such as the G2 method by Hoffman and Johnson [7]. In this study, six different numerical schemes were tested on six different test problems (making a total of 36 test cases). The test cases all involved laminar flow at small to moderate size Reynolds numbers in the range 1–1000. For each test problem, convergence in a functional of interest or a global error norm was studied for a sequence of refined meshes. The main conclusion of this study was that the IPCS scheme was, overall, the most accurate and efficient method for the particular choice of test problems. Based on these results, we choose to base our implementation on the IPCS scheme, but emphasize that the analysis is equally valid for any other scheme.

We consider the IPCS scheme in combination with a Taylor–Hood [14] approximation of the velocity and pressure variables; that is, we seek a solution $\bar{U} = (U, P) \in V_h \times Q_h$, where V_h is the space of continuous piecewise (vector-valued) quadratic polynomials and Q_h is the space of continuous piecewise linear polynomials, respectively. A summary of the IPCS scheme is given in Algorithm 1.

To analyze the error of the splitting method, one must construct a suitable interpolant/continuous extension in order for the solution to be defined at each point $(x, t) \in \Omega \times [0, T]$. Such an interpolant comes natural for the scheme under consideration. Since the solution is computed using a finite element formulation in space, we only need to consider interpolation in time. In time, we define the discrete solution to be the piecewise linear interpolant on each interval I_n based on the values U^{n-1}, U^n and P^{n-1}, P^n , respectively. For a higher order splitting scheme, care must be taken in the construction of the interpolant to maintain the order of accuracy.

4. *A posteriori* error analysis

To prove an *a posteriori* error estimate for the approximate solution of (1) computed by the inconsistent finite element formulation (the splitting scheme), we first state the weak form of (1) in Section 4.1 and the corresponding weak dual problem in Section 4.2. We then derive an error representation in Section 4.3 from which we obtain the error estimate(s) in Section 4.4.

4.1. The weak primal problem

The weak form of (1) reads: find $(u, p) \in W = V \times Q = \{v \in L^2(0, T; [H^1(\Omega)]^d) : v(\cdot, 0) = u^0, v|_{\Gamma_D} = 0\} \times \{q \in L^2(\Omega) : \langle q, 1 \rangle = 0\}$ such that

$$a((u, p); (v, q)) = L((v, q)) \quad (7)$$

for all $(v, q) \in \hat{W}$. The test space \hat{W} is defined analogously to the trial space with homogeneous initial conditions.

Algorithm 1 The Incremental Pressure Correction Scheme (IPCS)

Let $k_n = t_n - t_{n-1}$ denote the time step and $I_n = (t_{n-1}, t_n]$ the corresponding time interval. Furthermore, let V_h and \hat{V}_h denote a pair of trial and test spaces on a domain Ω . For each time interval I_n , we seek the fluid velocity $U^n = U(\cdot, t_n) \in V_h$ and pressure $P^n = P(\cdot, t_n) \in Q_h$ at time t_n by solving the following three variational problems:

- 1) Compute the tentative velocity U^\star by solving

$$\begin{aligned} \langle d_t^n(U^\star), v \rangle + \langle \sigma(U^{n-\frac{1}{2}}, P^{n-1}), \epsilon(v) \rangle \\ - \langle \nu(\nabla U^{n-\frac{1}{2}})^\top n, v \rangle_{\Gamma_N} + \langle P^{n-1} n, v \rangle_{\Gamma_N} = \langle f, v \rangle \end{aligned} \quad (4)$$

for all $v \in \hat{V}_h$, including any boundary conditions for the velocity. Here, $d_t^n(U^\star) = (U^\star - U^{n-1})/k_n + (U^{n-1} \cdot \nabla)U^{n-1}$ and $U^{n-\frac{1}{2}} = (U^\star + U^{n-1})/2$.

- 2) Compute the corrected pressure P^n by solving

$$\langle \nabla P^n, \nabla q \rangle = \langle \nabla P^{n-1}, \nabla q \rangle - k_n^{-1} \langle \nabla \cdot U^\star, q \rangle \quad (5)$$

for all $q \in \hat{Q}_h$, including any boundary conditions for the pressure.

- 3) Compute the corrected velocity U^n by solving

$$\langle U^n, v \rangle = \langle U^\star, v \rangle - k_n \langle \nabla(P^n - P^{n-1}), v \rangle \quad (6)$$

for all $v \in \hat{V}_h$, including any boundary conditions for the velocity.

The nonlinear form $a(\cdot; \cdot)$ and the linear form $L(\cdot)$ in (7) are defined as

$$\begin{aligned} a((u, p); (v, q)) &= \int_0^T \langle \dot{u} + (u \cdot \nabla)u, v \rangle + \langle \sigma(u, p), \epsilon(v) \rangle \\ &\quad - \langle \nu(\nabla u)^\top n - pn, v \rangle_{\Gamma_N} + \langle \nabla \cdot u, q \rangle \, dt, \end{aligned} \quad (8)$$

$$L((v, q)) = \int_0^T \langle f, v \rangle \, dt. \quad (9)$$

We let $r : \hat{W} \rightarrow \mathbb{R}$ denote the weak residual of (7); that is

$$r((v, q)) = L((v, q)) - a((u, p); (v, q)) = \int_0^T r^t((v, q)) \, dt \quad (10)$$

for all $(v, q) \in \hat{W}$.

4.2. The weak dual problem

Let now $\tilde{u} = (u, p)$ and let $\mathcal{M} = \mathcal{M}(\tilde{u})$ denote a given linear goal functional (the quantity of interest). The goal functional \mathcal{M} is assumed to be of the form

$$\mathcal{M}(\tilde{u}) = \mathcal{M}^T(u(\cdot, T)) + \int_0^T \mathcal{M}^t(\tilde{u}(\cdot, t)) \, dt. \quad (11)$$

Here, \mathcal{M}^T and \mathcal{M}^t describe target functionals at the end time $t = T$ and target functionals integrated over the time interval $[0, T]$, respectively.

We may now introduce the weak dual problem of the incompressible Navier–Stokes equations. The dual problem is used below in Section 4.3 to express the error in the goal functional \mathcal{M} in terms of the weak residual (10). We let $\tilde{z} = (z, y)$ denote the dual solution, where z is the *dual velocity* and y the *dual pressure*. The (abstract) weak dual problem reads: find $\tilde{z} \in W^*$ such that

$$\overline{a'}^*(\tilde{z}, \tilde{v}) = \mathcal{M}(\tilde{v}) \quad (12)$$

for all $\tilde{v} \in \hat{W}^*$. The dual trial and test spaces are defined by $(W^*, \hat{W}^*) = (\hat{W}, W_0)$ where $W_0 = \{v - w : v, w \in W\}$. The linearized, averaged and adjoint form $\overline{a'}^* : W^* \times \hat{W}^* \rightarrow \mathbb{R}$ is defined by

$$\overline{a'}^*(\tilde{v}, \delta \tilde{u}) = \overline{a'}(\delta \tilde{u}, \tilde{v}) = \int_0^1 a'(s\tilde{u} + (1-s)\tilde{U}; \tilde{v}) \delta \tilde{u} \, ds, \quad (13)$$

where a' denotes the Fréchet derivative of the nonlinear form $a : W \times \hat{W} \rightarrow \mathbb{R}$ with respect to its first argument.

To express the weak dual problem for the incompressible Navier–Stokes equations, we start from the abstract dual problem (12) and differentiate the nonlinear form a defined in (8) with respect to the velocity field u and the pressure field p . The adjoint operator * amounts to replacing the test functions (v, q)

in (8) with the dual variables (z, y) , and replacing the linearization variables $\delta \tilde{u} = (\delta u, \delta p)$ in (13) by the dual test functions (v, q) . We find that the dual variational problem reads: find $(z, y) \in W^*$ such that

$$\begin{aligned} \int_0^T \langle z, \dot{v} \rangle + \langle z, (\bar{u} \cdot \nabla)v + (v \cdot \nabla)\bar{u} \rangle + \langle \epsilon(z), \sigma(v, q) \rangle \\ - \langle z, \nu(\nabla v)^\top n - qn \rangle_{\Gamma_N} + \langle y, \nabla \cdot v \rangle \, dt \\ = \mathcal{M}^T(v(\cdot, T)) + \int_0^T \mathcal{M}^t((v, q)) \, dt \end{aligned} \quad (14)$$

for all $(v, q) \in \hat{W}^*$, where $\bar{u} = \frac{1}{2}(U + u)$. To solve (14), we integrate the first term by parts:

$$\int_0^T \langle z, \dot{v} \rangle \, dt = \int_0^T \langle -\dot{z}, v \rangle \, dt + \langle z(\cdot, T), v(\cdot, T) \rangle - \langle z(\cdot, 0), v(\cdot, 0) \rangle. \quad (15)$$

The boundary term at $t = 0$ vanishes since $(v, q) \in \hat{W}^* = W_0$ and thus $v(\cdot, 0) = 0$. The second term cancels the term $\mathcal{M}^T(v(\cdot, T))$ in the right-hand side of (14) if we take $z(\cdot, T) = \psi^T$ where ψ^T is the (L^2) Riesz representer of \mathcal{M}^T . It follows that the dual solution may be computed by solving the backward initial value problem

$$\begin{aligned} \int_0^T \langle -\dot{z}, v \rangle + \langle z, (\bar{u} \cdot \nabla)v + (v \cdot \nabla)\bar{u} \rangle + \langle \epsilon(z), \sigma(v, q) \rangle \\ - \langle z, \nu(\nabla v)^\top n - qn \rangle_{\Gamma_N} + \langle y, \nabla \cdot v \rangle \, dt \\ = \int_0^T \mathcal{M}^t((v, q)) \, dt, \end{aligned} \quad (16)$$

with initial condition $z(\cdot, T) = \psi^T$.

Remark 1. The dual solution may be computed by a direct finite element discretization of (16) with $\bar{u} \approx U$. In particular, it is not necessary to integrate by parts the remaining terms of (16) to move derivatives from the test function v .

4.3. Error representation

To derive a representation of the error $\mathcal{M}(\tilde{u}) - \mathcal{M}(\tilde{U}) = \mathcal{M}(\tilde{e})$ in terms of the solution $\tilde{z} = (z, y)$ of the dual problem (12) and the weak residual r defined in (10), we note that by the definition of the averaged linearized operator $\overline{a'}$ in (13), it follows that

$$\begin{aligned} \overline{a'}(\tilde{e}, \tilde{v}) &= \int_0^1 a'(s\tilde{u} + (1-s)\tilde{U}; \tilde{v})\tilde{e} \, ds = \int_0^1 \frac{d}{ds} a(s\tilde{u} + (1-s)\tilde{U}; \tilde{v}) \, ds \\ &= a(\tilde{u}; \tilde{v}) - a(\tilde{U}; \tilde{v}), \end{aligned} \quad (17)$$

for all $\tilde{v} \in W^*$, where $\tilde{e} = \tilde{u} - \tilde{U} \in \hat{W}^*$. The error representation now follows directly by taking $\tilde{v} = \tilde{e} \in \hat{W}^*$ in (12):

$$\mathcal{M}(\tilde{e}) = \overline{a}^*(\tilde{z}, \tilde{e}) = \overline{a}'(\tilde{e}, \tilde{z}) = a(\tilde{u}; \tilde{z}) - a(\tilde{U}; \tilde{z}) = L(\tilde{z}) - a(\tilde{u}; \tilde{z}) = r(\tilde{z}). \quad (18)$$

In other words, the error in the goal function \mathcal{M} is the (weak) residual of the dual solution.

If now the solution \tilde{U} satisfies the Galerkin orthogonality $r(\tilde{v}) = 0$ for all $\tilde{v} \in \hat{W}_{hk} \subset \hat{W}$, one may subtract a test space interpolant $\pi_{hk}\tilde{z}$ to obtain $\mathcal{M}(\tilde{e}) = r(\tilde{z} - \pi_{hk}\tilde{z})$ from which the error estimate follows; see Eriksson and Johnson [15], Becker and Rannacher [2], Bangerth and Rannacher [16]. However, if the solution does not satisfy the Galerkin orthogonality, one must account for the lack of orthogonality by adding and subtracting the orthogonality condition. We do this in two steps to separately account for the effects of space discretization, time discretization and lack of orthogonality:

$$\begin{aligned} \eta &\equiv \mathcal{M}(\tilde{e}) = r(\tilde{z}) \\ &= r(\tilde{z} - \pi_h\tilde{z} + \pi_h\tilde{z} - \pi_{hk}\tilde{z} + \pi_{hk}\tilde{z}) \\ &= r(\tilde{z} - \pi_h\tilde{z}) + r(\pi_h\tilde{z} - \pi_{hk}\tilde{z}) + r(\pi_{hk}\tilde{z}) \\ &\equiv \eta_h + \eta_k + \eta_c. \end{aligned} \quad (19)$$

Here, π_h is an interpolant into the semi-discrete space of continuous piecewise quadratic vector-valued velocity fields and continuous piecewise linear scalar pressure fields at each time $t \in [0, T]$, π_k is an interpolant into the semi-discrete space of discontinuous piecewise constant functions (at each point $x \in \Omega$), and $\pi_{hk} = \pi_k\pi_h$ is an interpolant into the fully discrete test space $\hat{W}_{hk} \subset \hat{W}$.

4.4. Error estimates

To construct an adaptive algorithm based on the error representation (19), we estimate η_h , η_k and η_c in terms of computable quantities to obtain the total error estimate

$$|\eta| = |\eta_h + \eta_k + \eta_c| \leq |\eta_h| + |\eta_k| + |\eta_c| \leq E_h + E_k + E_c \equiv E, \quad (20)$$

where $|\eta_h| \leq E_h$, $|\eta_k| \leq E_k$ and $|\eta_c| \leq E_c$.

4.4.1. The space discretization error estimate E_h

Starting from the definition $\eta_h = r(\tilde{z} - \pi_h\tilde{z})$, we integrate by parts on each cell $K \in \mathcal{T}$, where \mathcal{T} denotes the triangulation of Ω , to obtain

$$\eta_h \leq \int_0^T \sum_{K \in \mathcal{T}} \eta_K \, dt \equiv E_h, \quad (21)$$

where $\eta_K = |\eta_K^1| + |\eta_K^2| + |\eta_K^3| + |\eta_K^4|$ and

$$\eta_K^1 = \langle \dot{U} + (U \cdot \nabla)U - \nabla \cdot \sigma(U, P) - f, z - \pi_h z \rangle_K, \quad (22)$$

$$\eta_K^2 = \langle \tfrac{1}{2}[\sigma(U, P)]_n, z - \pi_h z \rangle_{\partial K \setminus \partial \Omega}, \quad (23)$$

$$\eta_K^3 = \langle \nu \nabla U \, n, z - \pi_h z \rangle_{\partial K \cap \Gamma_N}, \quad (24)$$

$$\eta_K^4 = \langle \nabla \cdot U, y - \pi_h y \rangle_K. \quad (25)$$

Here, $\llbracket \sigma(U, P) \rrbracket_n = \sigma(U^+, P^+) n^+ + \sigma(U^-, P^-) n^-$ denotes the jump of the discrete normal stress $\sigma(U, P) n$ across (interior) edges ∂K . The time integral in (21) is evaluated using midpoint quadrature on each time interval I_n .

In practice, we approximate the dual solution \tilde{z} by a numerical approximation \tilde{Z} . However, care must be taken when inserting the approximation \tilde{Z} into the error representation (19) or the error estimate (21). In particular, the error representation will evaluate to zero if the primal solution satisfies the Galerkin orthogonality and the approximate dual solution is computed on the same mesh and using the same order as the primal solution. Furthermore, the error estimate will evaluate to zero since $\tilde{Z} - \pi_h \tilde{Z} = \tilde{Z} - \tilde{Z} = 0$. Instead, we compute an enhanced version $\mathcal{E}_h \tilde{Z}$ from the computed dual solution \tilde{Z} by local extrapolation on patches, as described in Rognes and Logg [17]. This allows the dual problem to be solved on the same mesh using the same order as the primal problem, which has many practical advantages.

4.4.2. The time discretization error estimate E_k

The time discretization error η_k is estimated by

$$\begin{aligned} |\eta_k| &= |r(\pi_h \tilde{z} - \pi_{hk} \tilde{z})| = \left| \int_0^T r^t(\pi_h \tilde{z} - \pi_{hk} \tilde{z}) \, dt \right| \leq \int_0^T |r^t(\pi_h \tilde{z} - \pi_{hk} \tilde{z})| \, dt \\ &\equiv E_k. \end{aligned} \tag{26}$$

To evaluate the estimate E_k , we face the problem of integrating the residual over each time interval I_n . This is challenging as the residual oscillates heavily on each interval. For time discretizations defined by a continuous Galerkin finite element method in time, the residual is orthogonal to a space of discontinuous piecewise polynomial functions on the partition of the time interval $[0, T]$. As a consequence, the residual behaves like a Legendre polynomial on each time interval [18]. This is not necessarily the case for a solution computed by a splitting method, as is the case here. However, for the sake of analysis, we make the assumption that the residual takes its maximum value at the endpoints of each interval I_n . For a piecewise linear finite element approximation in time, the corresponding test space consists of the space of discontinuous piecewise constant functions. We may then take the interpolant π_k to be the midpoint value on each interval to obtain the estimate

$$\begin{aligned} E_k &\leq \sum_{n=1}^M k_n |r^t(\tilde{Z}(\cdot, t_n)) - r^t((\tilde{Z}(\cdot, t_{n-1}) + \tilde{Z}(\cdot, t_n))/2)| \\ &= \frac{1}{2} \sum_{n=1}^M k_n |r^t(\tilde{Z}(\cdot, t_n)) - r^t(\tilde{Z}(\cdot, t_{n-1}))|, \end{aligned} \tag{27}$$

where \tilde{Z} is the approximate numerical solution of the dual problem and M is the number of time steps.

The estimate (27) is used to estimate the size of the time discretization error η_k by the adaptive algorithm presented below in Section 5. To control the size

of the adaptive time steps, we here derive an alternate estimate. We let R^t denote the L^2 Riesz representer of the functional r^t and write

$$\begin{aligned}
E_k &= \int_0^T |r^t(\pi_h \tilde{z} - \pi_{hk} \tilde{z})| \, dt = \int_0^T |\langle\langle \pi_h \tilde{z} - \pi_{hk} \tilde{z}, R^t \rangle\rangle| \, dt \\
&\leq \int_0^T \|\pi_h \tilde{z} - \pi_{hk} \tilde{z}\| \|R^t\| \, dt \\
&\leq \max_{[0,T]} \{k_n(t) \|R^t\|\} \int_0^T k_n^{-1} \|\pi_h \tilde{z} - \pi_{hk} \tilde{z}\| \, dt \\
&= S(T) \max_{[0,T]} \{k_n(t) \|R^t\|\} \\
&\equiv \bar{E}_k,
\end{aligned} \tag{28}$$

where $S(T) = \int_0^T k_n^{-1} \|\pi_h \tilde{z} - \pi_{hk} \tilde{z}\| \, dt$ is a stability factor. The inner product $\langle\langle \cdot, \cdot \rangle\rangle$ is here defined by $\langle\langle \tilde{u}, \tilde{v} \rangle\rangle = \langle u, v \rangle + \langle p, q \rangle$ and the norm $\|\cdot\|$ is defined as the corresponding norm. We remark that the introduction of inequalities in (28) may render the estimate less sharp. This however is not a problem since (28) is not used to estimate the error; it is only used to drive the selection of adaptive time steps.

The norm $\|R^t\|$ of the Riesz representer may be computed directly as follows. We first note that the Riesz representer R^t is defined by the variational problem

$$\langle\langle R^t, \tilde{v} \rangle\rangle = r^t(\tilde{v}) \tag{29}$$

for all test functions $\tilde{v} \in \hat{W}_h$. The variational problem (29) corresponds to a linear system

$$M\mathcal{R} = b \tag{30}$$

where M is the mass matrix and \mathcal{R} is the vector of degrees of freedom for R^t . Clearly, the solution to (30) is given by $\mathcal{R} = M^{-1}b$. It follows that

$$\begin{aligned}
\|R^t\|^2 &= \langle\langle R^t, R^t \rangle\rangle = \langle\langle \sum_{i=1}^N \mathcal{R}_i \varphi_i, \sum_{j=1}^N \mathcal{R}_j \varphi_j \rangle\rangle = \sum_{i,j=1}^N \mathcal{R}_i \langle\langle \varphi_i, \varphi_j \rangle\rangle \mathcal{R}_j \\
&= (M^{-1}b)^\top M \mathcal{R} = b^\top \mathcal{R},
\end{aligned} \tag{31}$$

by the symmetry of M . The residual norm $\|R^t\|$ may thus be computed by assembling and solving the linear system (30), computing the inner product $b^\top \mathcal{R}$ and taking the square root.

4.4.3. The computational error estimate E_c

The computational error η_c is computed by a direct evaluation of the weak residual for the computed approximate dual solution \tilde{Z} :

$$|\eta_c| = |r(\pi_{hk} \tilde{z})| \approx |r(\pi_k \tilde{Z})| = \left| \int_0^T r^t(\pi_k \tilde{Z}) \, dt \right| \equiv E_c, \tag{32}$$

where \tilde{Z} is the approximate solution of the dual problem computed on the same mesh and using the same polynomial degree and time steps as the primal solution. In our implementation, we have chosen to compute the dual solution by a simple application of the dG(0) (backward Euler) method to the linear dual problem. We note that, by construction, the computational error estimate E_c is zero if the primal solution satisfies the Galerkin orthogonality.

5. Adaptive algorithm

Based on the *a posteriori* error estimate derived in Section 4.4, we may now formulate an adaptive algorithm for the incompressible Navier–Stokes equations. The adaptive algorithm is summarized in Algorithm 2.

Algorithm 2 Adaptive algorithm

Given a goal functional $\mathcal{M} = \mathcal{M}(\tilde{u})$ and a tolerance $\text{TOL} > 0$:

- 0) Select an initial coarse mesh and initial time step.
 - 1) Solve the primal problem (1) using (for example) the incremental pressure correction scheme (Algorithm 1) on the current (fixed) mesh using adaptive time steps.
 - 2) Solve the dual problem (16) backward in time on the same mesh as the primal problem and using the same adaptive time steps.
 - 3) Evaluate the error estimate $E = E_h + E_k + E_c$ defined in (21), (27) and (32), and the error indicators $\{\eta_K\}$.
 - 4) If $E \leq \text{TOL}$, then stop.
 - 5) Refine the mesh based on the error indicators $\{\eta_K\}$.
 - 6) Continue from step 1).
-

In Algorithm 2, we make use of two different tolerances TOL_h and TOL_k which are used to control the errors in the space and time discretization such that $\text{TOL}_h + \text{TOL}_k \leq \text{TOL} - E_c$. The computational error E_c is only used as part of the stopping criterion $E \leq \text{TOL}$; it is not used to drive the adaptive refinement. However, as will be demonstrated in Section 6, the computational error is reduced when the size of the time step is reduced. One may therefore consider extending the adaptive algorithm to control also the size of the computational error E_c .

In each adaptive iteration, consisting of a full solution of the primal problem, the dual problem and evaluation of the error indicators, the mesh is adaptively refined based on fixed fraction marking; that is, a fixed top fraction of the cells with the largest indicators are marked for refinement. For mesh refinement, we consider two different refinement strategies: the Rivara recursive bisection algorithm [19] and a regular cut algorithm which subdivides all marked triangles into four congruent subtriangles and propagates the refinement to neighboring triangles using bisection.

The step size k_n is determined in each time step based on the error estimate $\bar{E}_k = S(T) \max_{[0,T]} \{k_n(t) \|R^t\|\}$. To achieve $\bar{E}_k = \text{TOL}_k$, we set

$$k_n = \frac{\text{TOL}_k}{S(T) \max_{[t_{n-1}, t_n]} \|R^t\|} = \frac{\text{TOL}_k}{S(T) \|R^n\|}, \quad (33)$$

where again we have made the assumption that the residual takes its maximum value at the endpoints. Since R^n is not known until the solution has been computed on the time interval I_n , which in turn depends on the size of the time step k_n , it is tempting to replace R^n by R^{n-1} in (33). However, this leads to oscillations in the time step; if R^{n-1} is large, k_n will be small and, as a consequence, R^n will be small, which in turn leads to a large step k_n and so on. To control the time step, one may introduce a form of smoothing by letting \tilde{k}_n be the time step determined by

$$\tilde{k}_n = \frac{\text{tol}_k}{\|R^{n-1}\|}, \quad (34)$$

for $\text{tol}_k = \text{TOL}_k/S(T)$ and then take k_n to be the harmonic mean

$$k_n = \frac{2k_{n-1}\tilde{k}_n}{k_{n-1} + \tilde{k}_n}. \quad (35)$$

See Söderlind [20] and Logg [21] for a further discussion on time step selection. In practice, we do not compute the stability factor $S(T)$ but instead adjust the size of tol_k based on the size of E_k .

6. Numerical results

We here present numerical results to test the adaptive algorithm and the quality of the derived error estimates. An implementation of the adaptive solver, including the test problems described in this section, is freely available as part of the open source solver package CBC.Solve [22]. The package relies on the FEniCS/DOLFIN finite element library [23, 24, 25].

6.1. Case 1: Channel flow with wall-mounted body

As a first test problem, we consider a wall-mounted body (a “flap”) immersed in a pressure-driven channel flow as illustrated in Figure 1. The kinematic viscosity is $\nu = 0.002$. As initial condition, we set $u = 0$. The pressure boundary conditions $p = 1$ at the inflow and $p = 0$ at the outflow accelerate the flow from the initial stationary (zero) solution to the flow field depicted in Figure 2 at final time $T = 2.5$. Note that the solution at final time is not stationary which is important when measuring the performance and propagation of time discretization errors.

As a goal functional, we consider the integrated shear stress on the top of the flap:

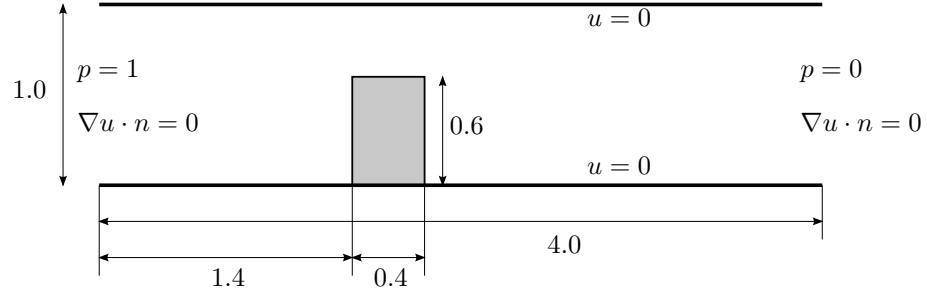


Figure 1: Geometry and boundary conditions for the “channel with flap” model problem.

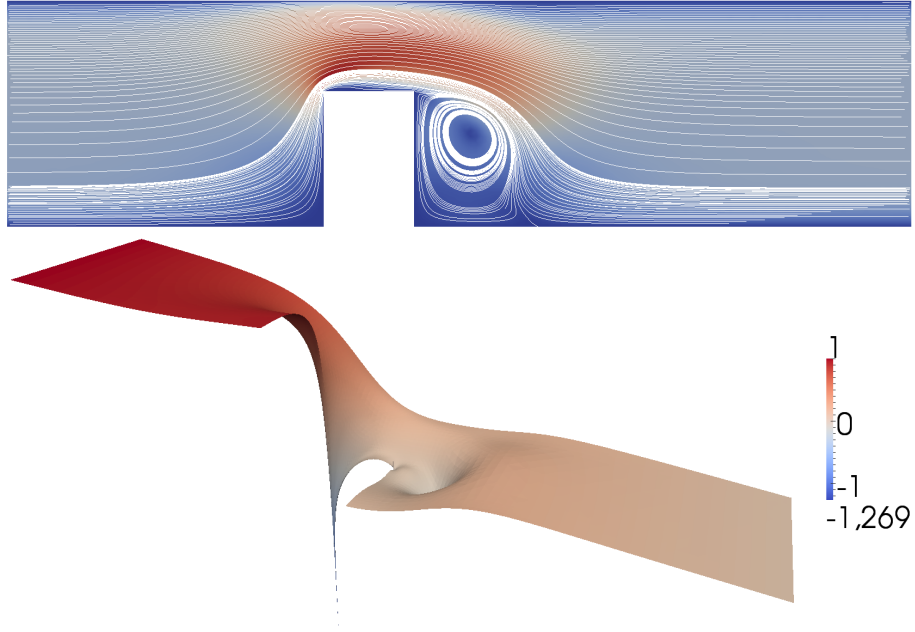


Figure 2: Fluid velocity (top) and pressure (bottom) at final time $T = 2.5$ computed with fixed time step $k = 0.005$ and 14 levels of bisection refinement (marking fraction 0.3). The final mesh has 16,581 triangles (76,085 degrees of freedom). The colorbar indicates the range of the scalar pressure field.

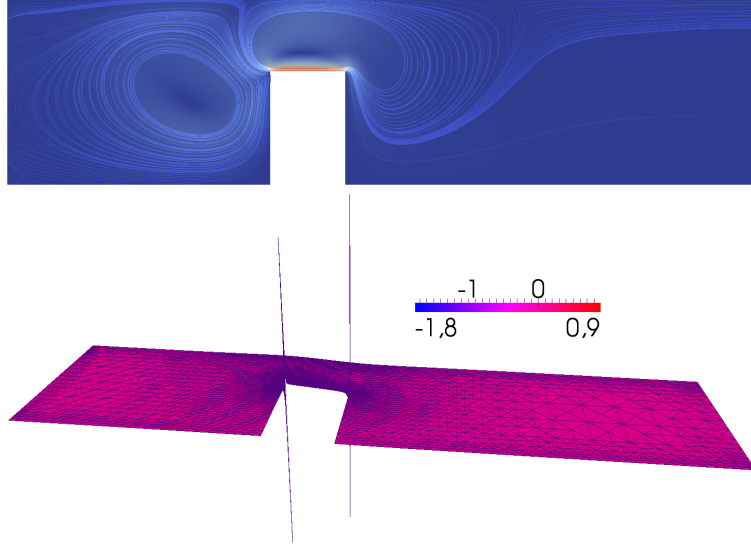


Figure 3: Dual fluid velocity (top) and dual pressure (bottom) at “final” time $t = 0$ for the channel flow test problem. The colorbar indicates the range of the scalar dual pressure field.

$$\begin{aligned}
\mathcal{M}_1(\tilde{u}) &= \int_0^T \int_{\Gamma_1} (\sigma(u, p) n) \cdot t \, ds \, dt \\
&= \int_0^T \int_{\Gamma_1} \sigma_{12}(u, p) \, ds \, dt = \int_0^T \int_{\Gamma_1} \nu (\partial u_1 / \partial x_2 + \partial u_2 / \partial x_1) \, ds \, dt,
\end{aligned} \tag{36}$$

where $n = (0, 1)$, $t = (1, 0)$ and $\Gamma_1 = \{(x_1, x_2) : 1.4 \leq x_1 \leq 1.8, x_2 = 0.6\}$. As a reference value for the goal functional, we take $\mathcal{M}_1(\tilde{u}) = 0.0200$. This reference value was obtained by extrapolation from solutions computed with constant time step $k = 0.005$ on a sequence of adaptively refined meshes.

6.1.1. Dual solutions and adaptive meshes

The dual solutions corresponding to the goal functional \mathcal{M}_1 are shown in Figure 3. As seen in this Figure, the dual solution clearly reflects the choice of goal functional. The dual velocity z (and dual velocity gradients) are large close to the top of the flap where the goal functional \mathcal{M}_1 measures the shear stress. The same figure displays large spikes in the dual pressure y at the reentrant corners. A detail of the dual pressure spikes is displayed in Figure 4. Combined with large residuals in the vicinity of the reentrant corners, the result is heavy refinement in a region located close to the top of the flap as seen in Figure 5.

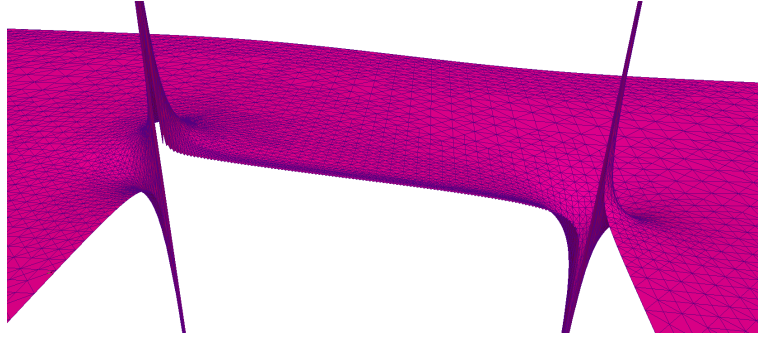


Figure 4: Detail of the dual pressure field at “final” time $t = 0$ for the channel flow test problem.

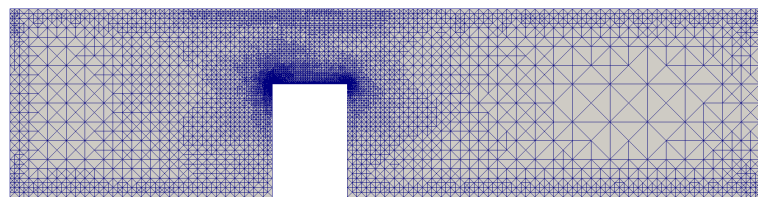


Figure 5: Mesh obtained by 14 levels of recursive bisection refinement with marking fraction 0.3 for the channel flow test problem.

6.1.2. Convergence and efficiency indices

To investigate the performance of adaptive mesh refinement and the quality of computed error estimates, we plot in Figure 6 errors and efficiency indices for a sequence of adaptively refined meshes and fixed time step $k = 0.005$. A comparison is made between three different refinement algorithms: recursive bisection, regular cut refinement and uniform (non-adaptive) refinement. For both recursive bisection and regular cut refinement, we use a fixed fraction marking strategy with marking fraction 0.3; that is, in each refinement step, we mark for refinement the top 30% of all triangles with the largest error indicators.

We find that the adaptive algorithm performs very well; a uniformly refined mesh requires more than an order of magnitude more degrees of freedom to reach a prescribed tolerance. This is evident in Figure 6 by finding the point where the error reaches the level $|\mathcal{M}_1(\tilde{e})| \leq 0.001$. This level is reached for roughly 90,000 degrees of freedom on a uniformly refined mesh, whereas the adaptively refined meshes obtained by recursive bisection and regular cut refinement reach the same level of accuracy using only 5,000 and 10,000 degrees of freedom, respectively. We also note that while the solution obtained by recursive bisection converges fastest, the convergence of the solution obtained by regular cut refinement is more robust. Computed efficiency indices (error estimate divided by actual error) are stable and vary between ca. 3 and 4, which means that we overestimate the error, but not by much.

To study the effect of the choice of marking fraction, we plot in Figure 7 errors and efficiency indices for marking fractions 0.1, 0.2, 0.3, 0.4 and 0.5 for fixed fraction bisection refinement. We note that while a smaller marking fraction gives rise to more efficient meshes, that is, a smaller number of degrees of freedom are needed to reach a given level of accuracy, more refinement levels are needed to reach that level of accuracy.

6.1.3. Convergence of the global adaptive algorithm

We next consider the convergence of the global adaptive algorithm. A tolerance $\text{TOL} = 0.001$ is prescribed for the error in the goal functional, here the shear stress goal functional \mathcal{M}_1 defined in (36), and ask the global adaptive algorithm described in Section 5 to adaptively refine the mesh and select adaptive time steps such that $|\mathcal{M}_1(\tilde{e})| < \text{TOL}$.

Figure 8 shows the convergence of the global adaptive algorithm. The algorithm converges in five iterations when the prescribed tolerance of $\text{TOL} = 0.001$ has been reached. Although the actual error reaches the prescribed tolerance after only three refinements, the adaptive algorithm performs well; the size of the efficiency index is ca. 3. The adaptive time steps are shown in Figure 9. At $t = 0$, the time step is set to the smallest time step from the previous refinement level. Since the solution is initially at rest, the time residual is initially small which leads to an increase in the size of the time steps. As the fluid is accelerated by the pressure gradient, the time residual increases and the time step is consequently reduced.

In Figures 10 and 11, we plot the different contributions to the total error estimate $E = E_h + E_k + E_c$. We find that the error is dominated by the space

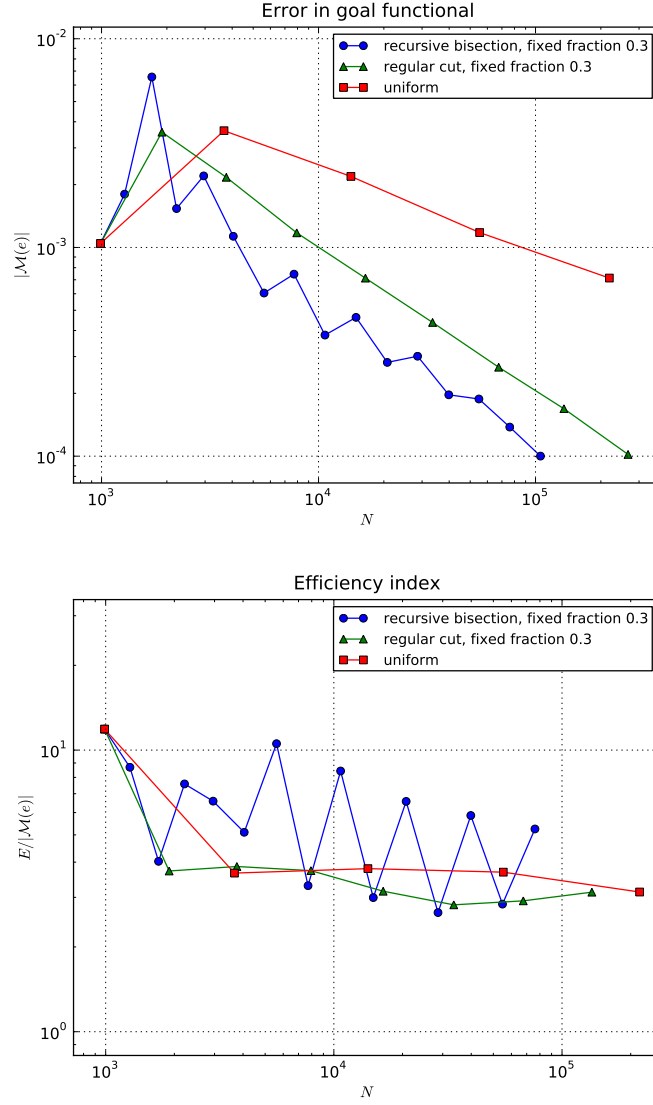


Figure 6: Error (top) and efficiency indices (bottom) as function of the number of spatial degrees of freedom for fixed time step $k = 0.005$, fixed fraction marking (marking fraction 0.3) and varying refinement algorithms (recursive bisection, regular cut and uniform) for the channel flow test problem.

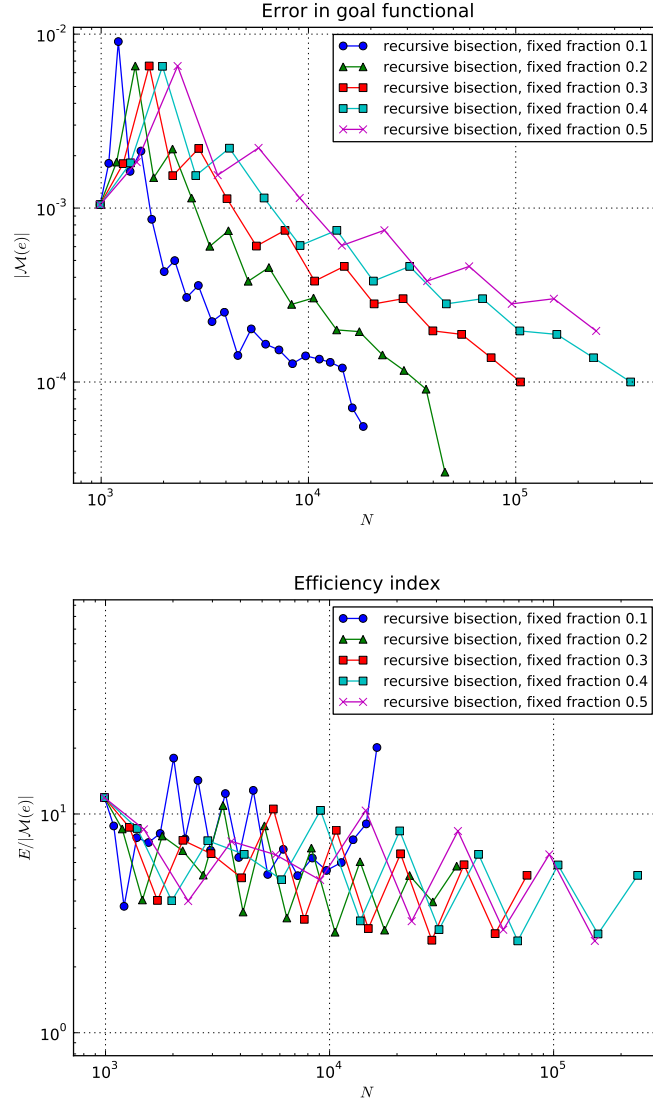


Figure 7: Comparison of errors (top) and efficiency indices (bottom) for varying marking fraction using fixed fraction bisection refinement for the channel flow test problem.

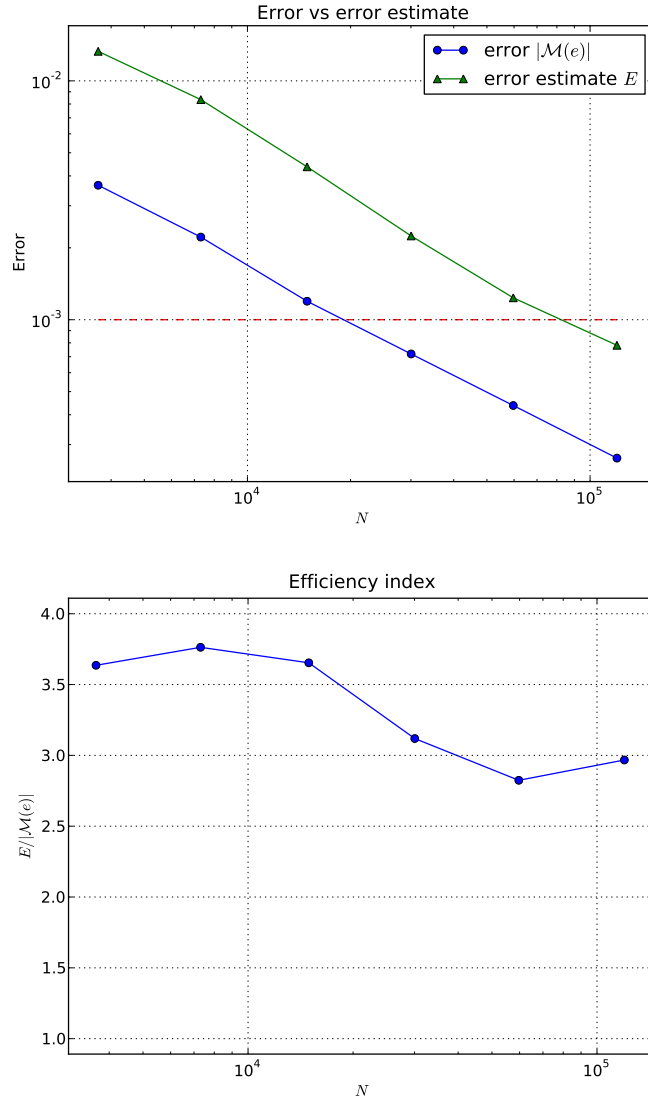


Figure 8: Convergence of the global space-time adaptive algorithm showing errors (top) and efficiency indices (bottom) using regular cut refinement with marking fraction 0.3 for the channel flow test problem. The given tolerance $TOL = 0.001$ is reached after five refinements.

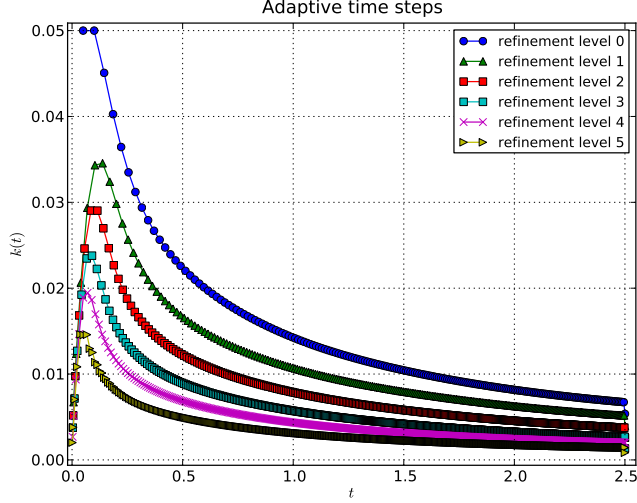


Figure 9: Time steps used by the global space–time adaptive algorithm on the time interval $[0, 2.5]$ for the channel flow test problem.

discretization error E_h , while the time discretization error remains small. This indicates that the time steps are unnecessarily small. However, the time steps must remain small to preserve stability of the numerical scheme. Although we have not taken any special measures to control the size of the time step to maintain numerical stability during mesh refinement, the adaptive time step selection seems to decrease naturally in each adaptive iteration as a result of an increase in the size of the residual $\|R\|$. The computational error E_c remains practically constant throughout the refinement and we note from Figure 11 that the dominating contribution to the computational error is from the momentum equation; the discrete residual of the continuity equation remains small.

6.1.4. Convergence as function of h and k

Finally, we investigate how the error contributions E_h , E_k and E_c depend on the mesh size h and the time step k . We consider the shear stress goal functional \mathcal{M}_1 defined in (36) computed on a sequence of uniformly refined meshes with mesh sizes $h = 0.2$, $h = 0.1$, $h = 0.05$ and $h = 0.025$, and fixed time steps $k = 0.01$, $k = 0.005$, $k = 0.0025$ and $k = 0.00125$.

Figure 12 shows the space discretization error E_h as function of the mesh size h . The results indicate that the convergence of the error in the goal functional is linear with respect to the mesh size. This has not been considered in detail but we note that for a P_2 – P_1 Taylor–Hood discretization, we expect the convergence of the error in the velocity to be h^3 in the mesh size. However, as the goal functional \mathcal{M}_1 involves the shear stress, the order of convergence is reduced to h^2 . The convergence rate is further decreased by the fact that the goal functional considers the shear stress on the boundary and as a result

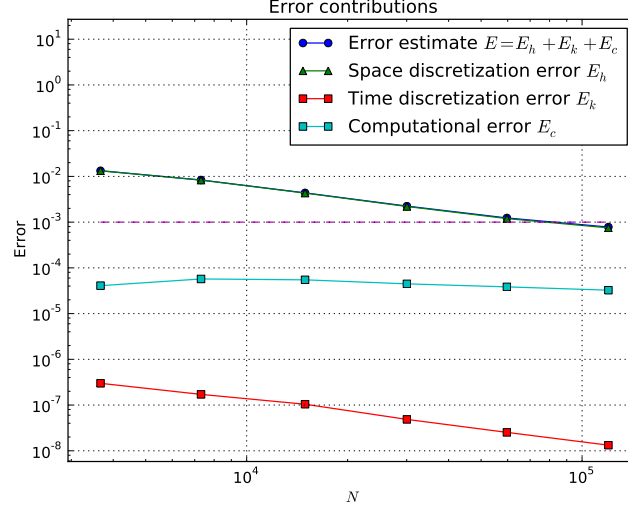


Figure 10: Contributions to the total error E from spatial discretization (E_h), time discretization (E_k) and computational (splitting) error (E_c) for the global space-time adaptive algorithm using regular cut refinement with marking fraction 0.3 for the channel flow test problem.

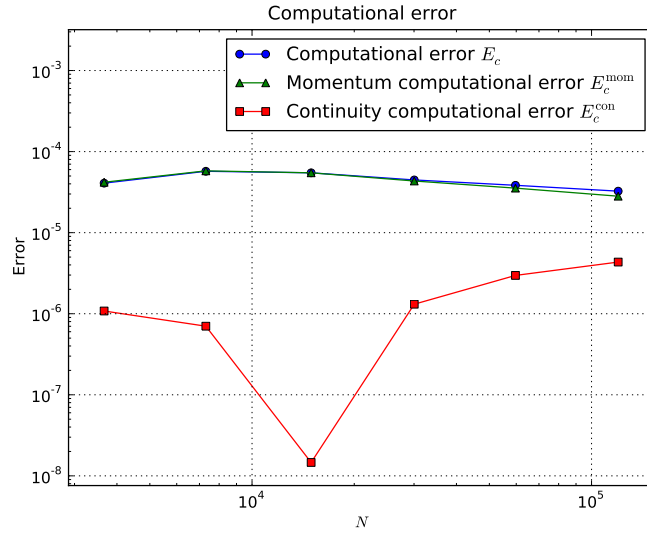


Figure 11: Contributions to the total computational (splitting) error E_c from inexact solution of the finite element formulation of the momentum equation (E_c^{mom}) and the continuity equation (E_c^{con}) using regular cut refinement with marking fraction 0.3 for the channel flow test problem.

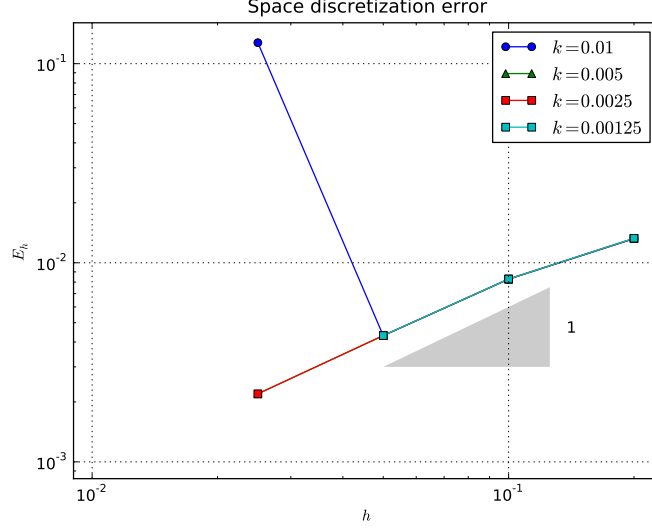


Figure 12: Space discretization error E_h for the shear stress goal functional \mathcal{M}_1 defined in (36) as function of mesh size h for varying (fixed) time steps.

of the singularities at the reentrant corners close to the evaluation of the goal functional. We further note from this figure that E_h does not depend on the size of the time step with one exception; the error goes up on the finest mesh for the largest time step $k = 0.01$, indicating instability of the numerical scheme for large relative time steps.

In Figure 13, we plot the time discretization error E_k as function of mesh size h and time step k , respectively. We conclude that E_k depends only weakly on h and that the convergence of E_k is quadratic in the time step k .

For the computational error E_c displayed in Figures 14 and 15, we similarly find a weak dependence on the mesh size h . We further note that the contribution from the momentum equation is linear in the size of the time step, whereas the contribution from the continuity equation is quadratic. Overall, we thus find that the order of convergence is linear in the time step as expected.

6.2. Case 2: Lid-driven cavity

As a second test problem, we consider the lid-driven cavity problem on the unit square $(0, 1) \times (0, 1)$. As boundary conditions, we set $u = (x_1(1 - x_1), 0)$ at the top of the cavity ($x_2 = 1$) with no-slip boundary conditions on the remaining boundary for the velocity. We also fix the pressure $p = 0$ at the bottom of the cavity ($x_2 = 0$). This “unphysical” boundary condition for the pressure gives rise to (small) gradients in the pressure field in the vicinity of $x_2 = 0$. A better way to ensure solvability of the pressure update step of Algorithm 1 is to require $\int_{\Omega} p \, dx = 0$. However, we have here chosen to use a Dirichlet boundary

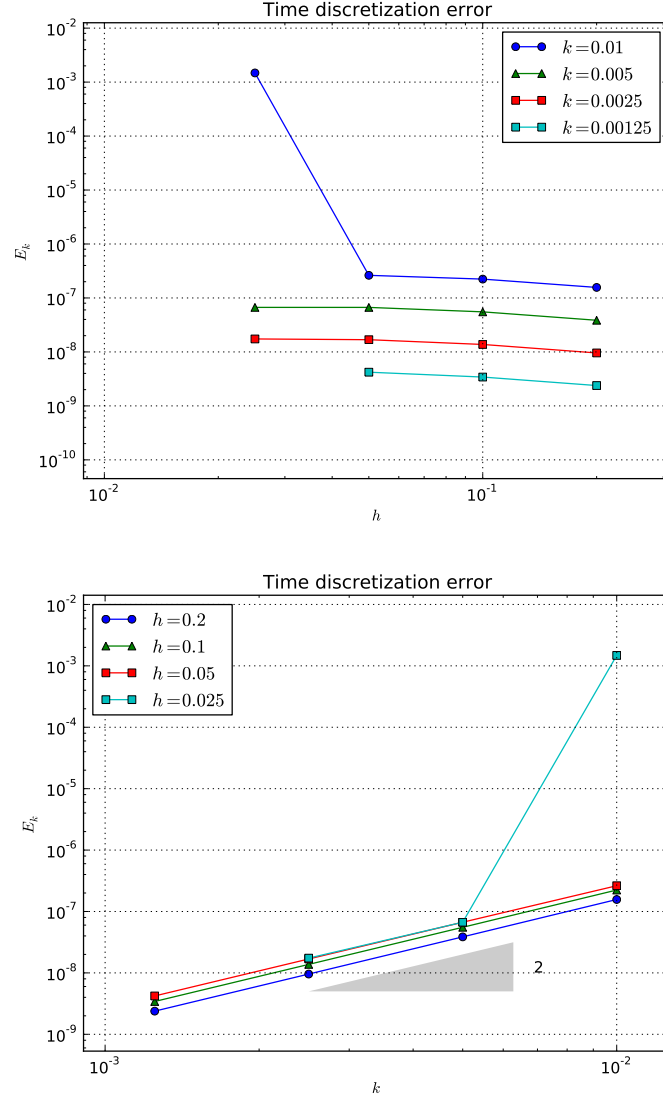


Figure 13: Time discretization error E_k for the shear stress goal functional \mathcal{M}_1 defined in (36) as function of mesh size h (top) and time step size k (bottom).

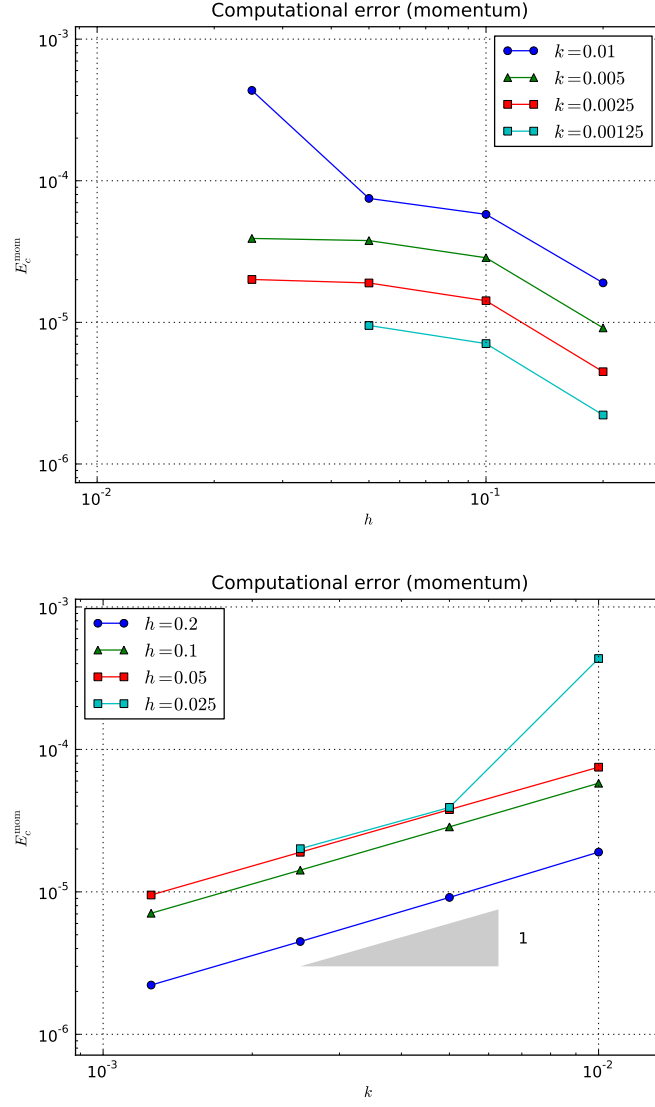


Figure 14: Momentum computational error E_C^{mom} for the shear stress goal functional \mathcal{M}_1 defined in (36) as function of mesh size h (top) and time step size k (bottom).

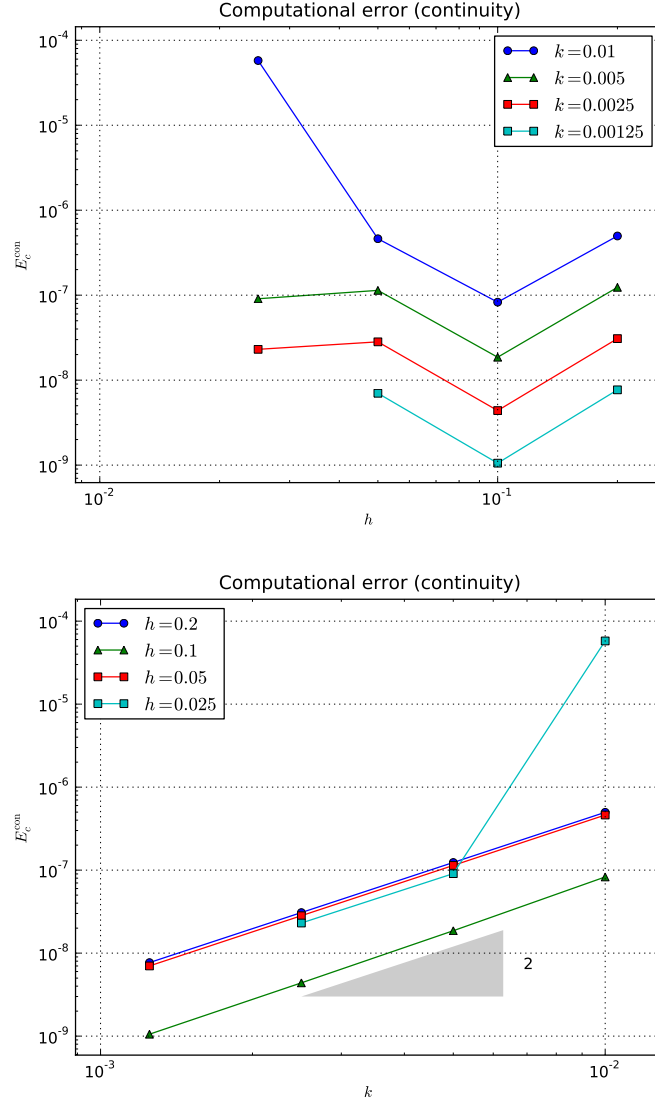


Figure 15: Continuity computational error E_C^{con} for the shear stress goal functional \mathcal{M}_1 defined in (36) as function of mesh size h (top) and time step size k (bottom).

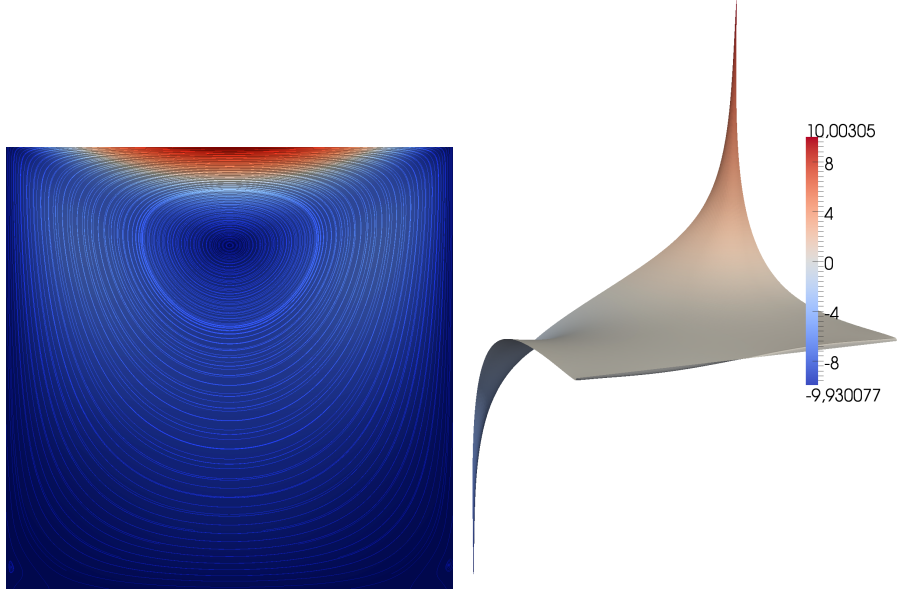


Figure 16: Velocity field (left) and pressure (right) at final time $T = 1$ for the lid-driven cavity test problem.

condition for the pressure, as this is often used in applications and we wish to study its effect on mesh refinement.

We set the kinematic viscosity to $\nu = 1$ and run the simulation over the time interval $[0, 1]$. As a goal functional, we consider a Gaussian-weighted average of the x_2 -component of the velocity field:

$$\mathcal{M}_2(\tilde{u}) = \int_0^T \int_{\Omega} u_2(x, t) \phi(x) \, dx \, dt. \quad (37)$$

The weight function ϕ is chosen as

$$\phi(x_1, x_2) = c \exp(-((x_1 - \bar{x}_1)^2 + (x_2 - \bar{x}_2)^2)/2r^2), \quad (38)$$

where $(\bar{x}_1, \bar{x}_2) = (0.75, 0.75)$, $r = 0.15$ and $c \approx 27.571034$ is chosen such that $\int_{\Omega} \phi(x_1, x_2) \, dx = 1$. As a reference value, we take $\mathcal{M}_2(\tilde{u}) = -0.039389$. The velocity and pressure fields at final time $T = 1$ are shown in Figure 16.

6.2.1. Dual solutions and adaptive meshes

The choice of goal functional generates a source located in the top right corner (at $x_1 = 0.75$). The dual solution is advected backwards along the primal velocity field and the resulting dual velocity field is shown in Figure 17. Notice the large secondary vortex in the top right corner and the small secondary vortices in the other three corners. The corresponding adaptive mesh is refined

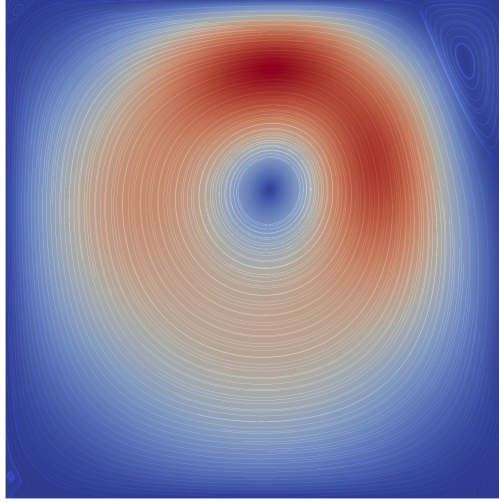


Figure 17: Dual fluid velocity field at “final” time $t = 0$ for the lid-driven cavity test problem.

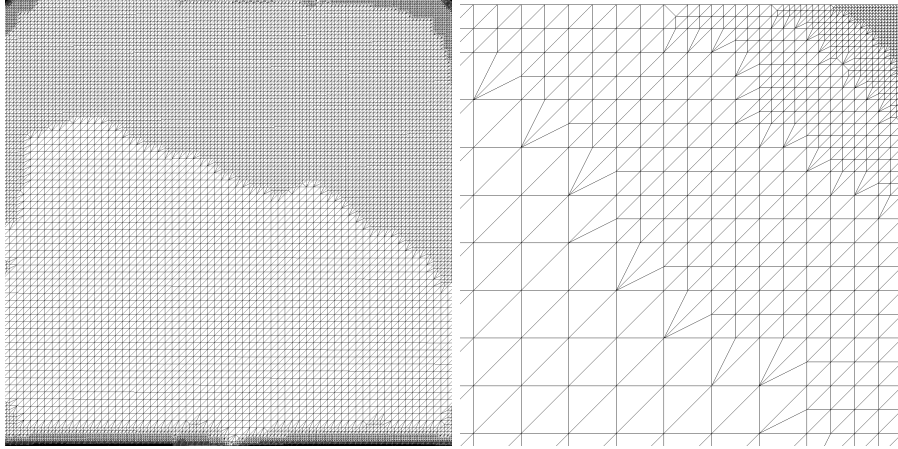


Figure 18: Mesh obtained by 12 levels of regular cut refinement with marking fraction 0.3 for the lid-driven cavity test problem (left) and a detailed view of the refined mesh in the top right corner (right).

heavily in the top left and right corners (see Figure 18), as a result of pressure spikes in these corners, but also at the bottom of the cavity as a result of the Dirichlet boundary condition used for the pressure.

6.2.2. Error and efficiency indices

Figure 19 shows the error of the goal functional and the corresponding efficiency indices for a sequence of adaptively refined meshes, using adaptive time-stepping on each refined mesh. Two different adaptive refinement algorithms, recursive bisection and regular cut refinement, are compared to uniform refinement. Both adaptive algorithms perform significantly better compared to uniform refinement. No significant difference can be noted between the two adaptive refinement algorithms, other than that recursive bisection requires approximately twice the number of refinement levels to reach the same level of accuracy as regular cut refinement. We further note that the efficiency indices vary between ca. 1 and 10. Interestingly, the efficiency indices for uniform refinement seem to converge towards 1.

7. Conclusions

We have presented an adaptive finite element method for the incompressible Navier–Stokes equations based on a standard splitting scheme (incremental pressure correction). By treating the splitting scheme as an approximation of a pure Galerkin finite element scheme, one may analyze the error as a sum of contributions from space discretization, time discretization and a computational error that measures the deviation of the splitting scheme from the pure Galerkin scheme. Numerical experiments indicate good performance of the adaptive algorithm and error estimates that closely match the actual error. The proposed method may thus serve as an attractive approach to solving the incompressible Navier–Stokes equations, combining the efficiency of a simple splitting method with the framework of goal-oriented adaptive finite element methods.

The presented adaptive algorithm can be further improved by extending the adaptive time step selection to control the size of the computational error E_c . It may also be interesting to consider modified splitting schemes to reduce the size of the computational error, in particular the size of the discrete momentum residual.

8. Acknowledgments

This work is supported by an Outstanding Young Investigator grant from the Research Council of Norway, NFR 180450. This work is also supported by a Center of Excellence grant from the Research Council of Norway to the Center for Biomedical Computing at Simula Research Laboratory.

References

- [1] K. Eriksson, D. Estep, P. Hansbo, C. Johnson, Introduction to adaptive methods for differential equations, *Acta numerica* 4 (1995) 105–158.
- [2] R. Becker, R. Rannacher, An optimal control approach to a posteriori error estimation in finite element methods, *Acta Numerica* 2001 10 (2001) 1–102.

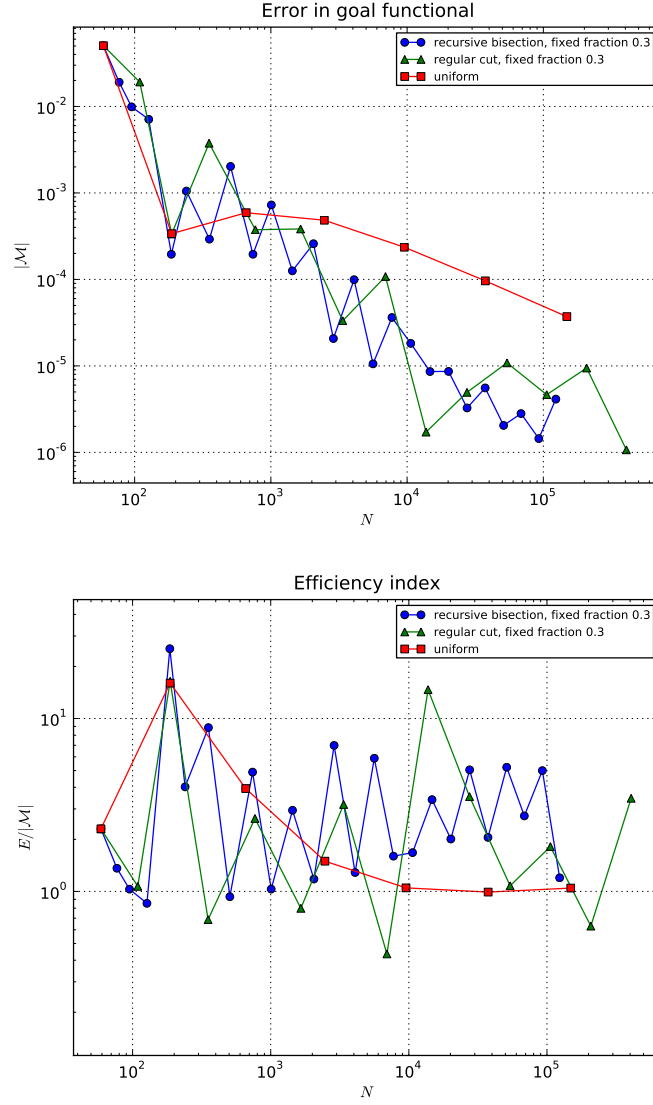


Figure 19: Error (top) and efficiency indices (bottom) as function of the number of spatial degrees of freedom using adaptive time steps, fixed fraction marking (marking fraction 0.3) and varying refinement algorithms (recursive bisection, regular cut and uniform) for the lid-driven cavity problem.

- [3] M. Giles, E. Süli, Adjoint methods for pdes: A posteriori error analysis and postprocessing by duality, *Acta Numerica* 11 (2002) 145–236.
- [4] D. Estep, M. Larson, R. Williams, Estimating the error of numerical solutions of systems of nonlinear reaction–diffusion equations, *Memoirs of the American Mathematical Society* 696 (2000) 1–109.
- [5] D. Estep, V. Ginting, D. Ropp, J. N. Shadid, S. Tavener, An a posteriori–a priori analysis of multiscale operator splitting, *SIAM Journal on Numerical Analysis* 46 (2008) 1116–1146.
- [6] J. Hoffman, On duality based a posteriori error estimation in various norms and linear functionals for LES, *SIAM J. Sci. Comput* 26 (2004) 178–195.
- [7] J. Hoffman, C. Johnson, *Computational Turbulent Incompressible Flow*, Springer, 2007.
- [8] A. Chorin, Numerical solution of the Navier–Stokes equations, *Mathematics of Computation* 22 (1968) 745–762.
- [9] R. Temam, Sur l’approximation de la solution des équations de Navier–Stokes par la méthode des pas fractionnaires (I), *Archive for Rational Mechanics and Analysis* 32 (1969) 135–153.
- [10] K. Goda, A multistep technique with implicit difference schemes for calculating two-or three-dimensional cavity flows, *Journal of Computational Physics* 30 (1979) 76–95.
- [11] F. Bengzon, M. G. Larson, A posteriori error estimates for fractional step methods in fluid mechanics, in: P. Bergan, J. Garcia, E. Onate, T. Kvamsdal (Eds.), *Computational methods in marine engineering*, CIMNE, Barcelona.
- [12] K. Valen-Sendstad, A. Logg, K.-A. Mardal, H. Narayanan, M. Mortensen, A comparison of finite element schemes for the incompressible Navier–Stokes equations, Chapter 21 in *Automated Solution of Differential Equations by the Finite Element Method*, Springer.
- [13] J. Guermond, J. Shen, A new class of truly consistent splitting schemes for incompressible flows, *Journal of Computational Physics* 192 (2003) 262–276.
- [14] C. Taylor, P. Hood, A numerical solution of the Navier-Stokes equations using the finite element technique, *Internat. J. Comput. & Fluids* 1 (1973) 73–100.
- [15] K. Eriksson, C. Johnson, Adaptive finite element methods for parabolic problems IV: Nonlinear problems, *SIAM Journal on Numerical Analysis* 32 (1995) 1729–1749.

- [16] W. Bangerth, R. Rannacher, Adaptive finite element methods for differential equations, Birkhäuser, 2003.
- [17] M. Rognes, A. Logg, Automated goal-oriented error control I: Stationary variational problems, SIAM Journal on Scientific Computing (in review) (2010).
- [18] A. Logg, Multi-adaptive Galerkin methods for ODEs I, SIAM J. Sci. Comput. 24 (2003) 1879–1902.
- [19] M. Rivara, Local modification of meshes for adaptive and/or multigrid finite-element methods, Journal of Computational and Applied Mathematics 36 (1991) 79–89.
- [20] G. Söderlind, Automatic control and adaptive time-stepping, Numerical Algorithms 31 (2002) 281–310.
- [21] A. Logg, Multi-adaptive Galerkin methods for ODEs II: Implementation and applications, SIAM J. Sci. Comput. 25 (2003) 1119–1141.
- [22] CBC.Solve, Software package, <http://www.launchpad.net/cbc.solve>.
- [23] FEniCS, FEniCS software collection, <http://www.fenicsproject.org>.
- [24] A. Logg, Automating the finite element method, Arch. Comput. Methods Eng. 14 (2007) 93–138.
- [25] A. Logg, G. N. Wells, DOLFIN: Automated finite element computing, ACM Transactions on Mathematical Software 32 (2010) 1–28.